anima
Internet-Draft
Intended status: Standards Track
Expires: May 2, 2017

S. Kumar
Philips Lighting Research
P. van der Stok
Consultant

October 29, 2016

EST based on DTLS secured CoAP (EST-coaps)
draft-vanderstok-core-coap-est-00

Abstract

Low-resource devices in a Low-power and Lossy Network (LLN) can operate in a mesh network using the IPv6 over Low-power Personal Area Networks (6LoWPAN) and IEEE 802.15.4 link-layer standards. Provisioning these devices in a secure manner with keys (often called security bootstrapping) used to encrypt and authenticate messages is the subject of Bootstrapping of Remote Secure Key Infrastructures (BRSKI) [I-D.ietf-anima-bootstrapping-keyinfra]. Enrollment over Secure Transport (EST) [RFC7030], based on TLS and HTTP, is used for BRSKI. This document defines how low-resource devices are expected to use EST over DTLS and CoAP. 6LoWPAN fragmentation management and minor extensions to CoAP are needed to enable EST over DTLS-secured CoAP (EST-coaps).

Note

Many of the concepts in this document are taken over from [RFC7030]. Consequently, much text is directly traceable to [RFC7030]. The same document structure is followed to point out the differences and commonalities between EST and EST-coaps.

Table of Contents

1. Introduction

IPv6 over Low-power Wireless Personal Area Networks (6LoWPANs) [RFC4944] on IEEE 802.15.4 [ieee802.15.4] wireless networks is becoming common in many professional application domains such as lighting controls. However commissioning of such networks suffers from a lack of standardized secure bootstrapping mechanisms for these networks.

Although IEEE 802.15.4 defines how security can be enabled between nodes within a single mesh network, it does not specify the provisioning and management of the keys. Therefore securing a 6LoWPAN network with devices from multiple manufacturers with

different provisioning techniques is often tedious and time consuming.

Bootstrapping of Remote Secure Infrastructures (BRSKI) [I-D.ietf-anima-bootstrapping-keyinfra] addresses the issue of bootstrapping networked devices in the context of Autonomic Networking Integrated Model and Approach (ANIMA). However, BRSKI has not been developed specifically for low-resource devices in constrained networks. These networks use DTLS [RFC6347], CoAP [RFC7252], and UDP instead of TLS [RFC5246], HTTP [RFC7230] and TCP. BRSKI relies on Enrollment over Secure Transport (EST) [RFC7030] for the provisioning of the operational domain certificates. Replacing the EST invocations of TLS and HTTP by DTLS and CoAP invocations enables applying BRSKI on CoAP-based low-resource devices.

The Figure 1 below shows the EST-coaps architecture.

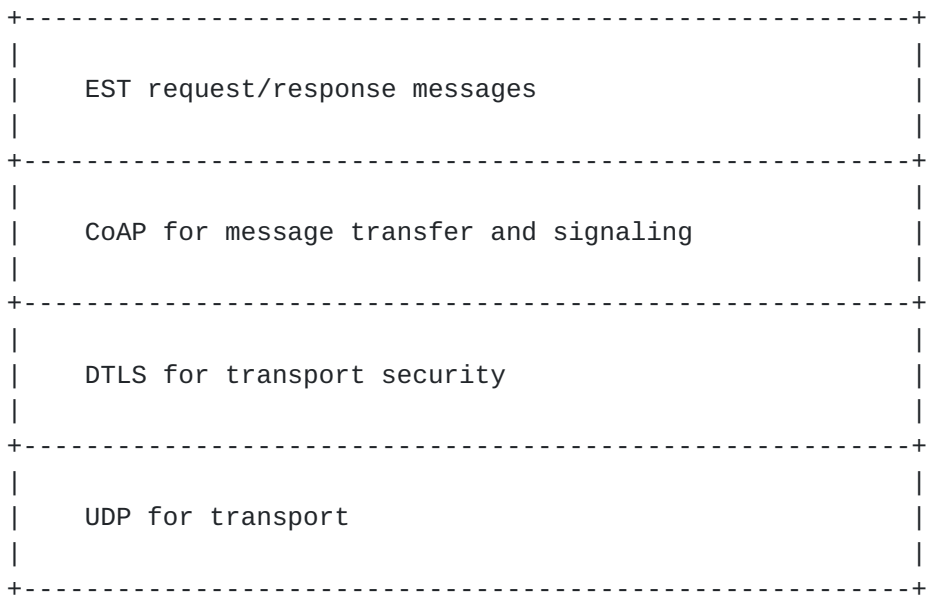


Figure 1: EST-coaps protocol layers

Although EST-coaps paves the way for the utilization of BRSKI for constrained devices on constrained networks, some devices will not have enough resources to handle the large payloads that come with EST-coaps. It is up to the network designer to decide which devices execute the BRSKI protocol and which not.

EST-coaps is designed for use in professional control networks such as lighting. The autonomic bootstrapping is interesting because it reduces the manual intervention during the commissioning of the

network. Typing in passwords is contrary to this wish. Therefore, the password authentication of EST is not supported in EST-coaps.

In the constrained devices context it is very unlikely that full PKI request messages will be used. For that reason, full PKI messages are not supported in EST-coaps.

Because the relatively large messages involved in EST cannot be readily transported over constrained (6LoWPAN, LLN) wireless networks, this document defines the use of CoAP Block-Wise Transfer ("Block") [RFC7959] combined with DTLS to fragment EST messages at the application layer.

1.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

All the terminology from EST [RFC7030] is included in this document by reference.

2. Operational Scenarios Overview

Only the differences to EST with respect to operational scenarios are described in this section. EST-coaps server authentication differs from EST as follows:

- Replacement of TLS by DTLS and HTTP by CoAP, resulting in:
  - DTLS-secured CoAP sessions between EST-coaps client and EST-coaps server.
- Only certificate-based client authentication is supported, with as result:
  - The EST-coaps client does not support manual authentication (as described in Section 4.4.1 of [RFC7030])
  - The EST-coaps client does not support authentication at the application layer.
- EST-coaps does not support full PKI request messages [RFC5272].

The following EST-coaps protocol parts are supported as described for the equivalent EST parts:

1. Request of client certificates by submitting a enrollment request to EST-coaps server.

2. Renewal of existing client certificates by submitting a re-enrollment request to EST-coaps server.

3. Request of certificate with key pair generated by EST-coaps server.

4. The EST-coaps client can request the attributes needed for enrollment before the enrollment request is issued"

3. Protocol Design and Layering

The EST-coaps protocol design follows closely the EST design, excluding some aspects that are not relevant for automatic bootstrapping of constrained devices within a professional context. The parts supported by EST-coaps are:

Message types:

* Simple PKI messages.

* CA certificate retrieval.

* CSR Attributes Request.

* Server-generated key request.

CoAP with Block-Wise Transfer:

* CoAP Block-Wise Transfer header Options for control of the transfer of larger EST messages.

DTLS for transport security:

* Authentication of the EST-coaps server.

* Authentication of the EST-coaps client.

* Communication integrity and confidentiality.

* Channel-binding information for linking proof-of-identity with message-based proof-of-possession (OPTIONAL).

Given that CoAP and DTLS can provide proof of identity for EST-coaps clients and server, simple PKI messages can be used conformant to section 3.1 of [RFC5272]. EST-coaps supports the certificate types

and Trust Anchors (TA) that are specified for EST in section 3 of [RFC7030].

The EST-coaps server URI is identical to the EST URI (except for replacing the scheme https by coaps):

coaps://www.example.com/.well-known/est
coaps://www.example.com/.well-known/est/arbitraryLabel1

See Figure 5 in section 3.2.2 of [RFC7030] for the path-suffixes (operations) that are supported by EST.

EST-coaps uses CoAP to transfer EST messages, aided by Block-Wise Transfer [RFC7959] to transport CoAP messages in blocks thus avoiding (excessive) 6LoWPAN fragmentation of UDP datagrams. The use of "Block" is specified in Section 3.2.

The content-format (media type equivalent) of the CoAP message determines which EST message is transported in the CoAP payload. The media types specified in the HTTP Content-Type header(see section 3.2.2 of [RFC7030]) are in EST-coaps specified by the Content-Format Option (12) of CoAP. The combination of URI path-suffix and content-format used MUST map to an allowed combination of path-suffix and media type as defined for EST.

EST-coaps is designed for use between low-resource devices using CoAP and hence does not need to send base64-encoded data. Simple binary coding is more efficient (30% less payload compared to base64) and well supported by CoAP. Therefore, the content formats specification in Section 5 requires the use of binary encoding for all EST-coaps CoAP payloads.

The functions of TLS specified for EST are in EST-coaps mapped to the equivalent DTLS functions. However, DTLS sessions SHOULD remain open for persistent EST-coaps connections to reduce storage load. For example, a cacerts request followed by an enrollments request SHOULD use the same DTLS session.

The mandatory cipher suite for DTLS is TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8 defined in [RFC7251] which is the mandatory-to-implement cipher suite in CoAP. Additionally the curve secp256r1 MUST be supported [RFC4492]; this curve is equivalent to the NIST P-256 curve. The hash algorithm is SHA-256. DTLS implementations MUST use the Supported Elliptic Curves and Supported Point Formats Extensions [RFC4492]; the uncompressed point format MUST be supported; [RFC6090] can be used as an implementation method.

3.1. CoAP response codes

Section 5.9 of [RFC7252] specifies the mapping of HTTP response codes to CoAP response codes. Every time the HTTP response code 200 is specified in [RFC7030] in response to a GET request, in EST-coaps the equivalent CoAP response code 2.05 MUST be used. Response code HTTP 202 in EST is mapped as indicated below; while other HTTP 2xx response codes are not used by EST. For the following HTTP 4xx error codes that may occur: 400, 401, 403, 404, 405, 406, 412, 413, 415 ; the equivalent CoAP response code for EST-coaps is 4.xx. For the HTTP 5xx error codes: 500, 501, 502, 503, 504 the equivalent CoAP response code is 5.xx.

HTTP response code 202 needs a different treatment from the one described for [RFC7030]. A new CoAP response code 2.06 is needed. When the EST over CoAP request cannot be treated immediately, a CoAP response code 2.06 Delayed is returned with Content-Format: application/link-format described in [RFC6690]. The payload of the response contains a link to receive the delayed response. ALTERNATIVE (to discuss) : a 2.06 Delayed response without payload and the link to receive the delayed response indicated using the Location-Path and Location-Query Options.

The waiting client may send GET requests to the returned link. When the response is not available, the server returns response code 2.06 with again the link for the client to query. When the response is available, the server returns the response code 2.05 Content with a payload containing the requested response in the appropriate content format.

3.2. Message fragmentation using Block

DTLS defines fragmentation only for the handshake part and not for secure data exchange (DTLS records). [RFC6347] states "Each DTLS record MUST fit within a single datagram". In order to avoid using IP fragmentation, which is not supported by 6LoWPAN, invokers of the DTLS record layer MUST size DTLS records so that they fit within any Path MTU estimates obtained from the record layer. In addition, invokers residing on a 6LoWPAN over IEEE 802.15.4 network SHOULD attempt to size CoAP messages such that each DTLS record will fit within one or two IEEE 802.15.4 frames only by choosing the appropriate block sizes.

Certificates can vary greatly in size dependent on signature algorithms and key sizes. For a 256-bit curve, common ECDSA sizes fluctuate between 500 bytes and 1 KB. Some EST messages may be several kilobytes in size. Given non-existence of IP fragmentation in 6LoWPAN networks and its 1280 bytes MTU, EST-coaps needs to be

able to fragment EST messages into multiple DTLS datagrams with each DTLS datagram containing a block of CoAP payload data. Further considering the small payload size available to a CoAP message, which can be as low as 68 bytes in case the message needs to fit into a single IEEE 802.15.4 frame, fine-grained fragmentation of EST messages is essential.

For CoAP, [RFC7959] specifies the "Block1" option for fragmentation of the request payload and the "Block2" option for fragmentation of the return payload. The CoAP client MAY specify the Block1 size and MAY also specify the Block2 size. The CoAP server MAY specify the Block2 size, but not the Block1 size.

Examples of fragmented messages are shown in Appendix A.

3.3. CoAP message headers

EST-coaps uses CoAP payload blocks that each fit in a single DTLS record i.e. UDP datagram without causing IP fragmentation. The returned CoAP response codes are specified in Section 3.1. The CoAP Token value is not specified by EST-coaps and may be chosen by the CoAP client according to [RFC7252].

An example HTTP request message cacerts in EST will look like:

```
REQ:
        GET /.well-known/est/cacerts HTTP/1.1
           Host: 192.0.2.1:8085
           Accept: */*

RES:
        HTTP/1.1 200 OK
        Status: 200 OK
        Content-Type: application/pkcs7-mime
        Content-Transfer-Encoding : base64
        Content-Length: 4246
        payload
```

The corresponding EST-coaps request looks like:

```
REQ:
        GET coaps://[192.0.2.1:8085]/.well-known/est/cacerts

RES:
        2.05 Content (Content-Format: application/pkcs7-mime)
        {payload}
```

4. Protocol Exchange Details

The EST-coaps client MUST be configured with an implicit TA database or an explicit TA database. The authentication of the EST-coaps server by the EST-coaps client is based on Certificate authentication in the DTLS handshake.

The authentication of the EST-coaps client is based on client certificate in the DTLS handshake. This can either be

- DTLS with a previously issued client certificate (e.g., an existing certificate issued by the EST CA);
- DTLS with a previously installed certificate (e.g., manufacturer-installed certificate or a certificate issued by some other party);

The details on checking the validity of the certificates are identical to EST.

The other protocol aspects such as simple enrollment (re-enrollment), certificate attributes and CA certificate request are similar to EST with the exception that these are performed on coaps (CoAP+DTLS) as the transport. The required content-formats for these request and response messages are defined in Section 5. The CoAP response codes are defined in Section 3.1.

EST-coaps does not support full PKI Requests. Consequently, the fullcmc request of section 4.3 of [RFC7030] and response MUST NOT be supported by EST-coaps.

5. IANA Considerations

Additions to the sub-registry "CoAP Content-Formats", within the "CoRE Parameters" registry are needed for the below media types. These can be registered either in the Expert Review range (0-255) or IETF Review range (256-9999).

1.

   * application/pkcs7-mime
   * Type name: application
   * Subtype name: pkcs7-mime
   * smime-type: certs-only

* ID: TBD1
* Required parameters: None
* Optional parameters: None
* Encoding considerations: Binary
* Security considerations: As defined in this specification
* Published specification: [RFC5751]
* Applications that use this media type: ANIMA Bootstrap (BRSKI) and EST

2.

* application/pkcs8
* Type name: application
* Subtype name: pkcs8
* ID: TBD2
* Required parameters: None
* Optional parameters: None
* Encoding considerations: Binary
* Security considerations: As defined in this specification
* Published specification: [RFC5958]
* Applications that use this media type: ANIMA Bootstrap (BRSKI) and EST

3.

* application/csrattrs
* Type name: application
* Subtype name: csrattrs
* ID: TBD3

* Required parameters: None

* Optional parameters: None

* Encoding considerations: Binary

* Security considerations: As defined in this specification

* Published specification: [RFC7030]

* Applications that use this media type: ANIMA Bootstrap (BRSKI) and EST

4.

* application/pkcs10

* Type name: application

* Subtype name: pkcs10

* ID: TBD4

* Required parameters: None

* Optional parameters: None

* Encoding considerations: binary

* Security considerations: As defined in this specification

* Published specification: [RFC5967]

* Applications that use this media type: ANIMA bootstrap (BRSKI) and EST

Additions to the sub-registry "CoAP Response Code", within the "CoRE Parameters" registry are needed for the following response codes:

o Code: 2.06

o Description: Delayed

o Reference: this document

6. Security Considerations

The security considerations mentioned in EST applies also to EST-coaps.

7. Acknowledgements

The authors are very grateful to Klaus Hartke for his detailed explanations on the use of Block with DTLS. The authors would like to thank Esko Dijk and Michael Verschoor for the valuable discussions that helped in shaping the solution.

8. Change Log

9. References

9.1. Normative References

[I-D.ietf-anima-bootstrapping-keyinfra]
Pritikin, M., Richardson, M., Behringer, M., and S. Bjarnason, "Bootstrapping Remote Secure Key Infrastructures (BRSKI)", draft-ietf-anima-bootstrapping-keyinfra-03 (work in progress), June 2016.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC4492] Blake-Wilson, S., Bolyard, N., Gupta, V., Hawk, C., and B. Moeller, "Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS)", RFC 4492, DOI 10.17487/RFC4492, May 2006, <http://www.rfc-editor.org/info/rfc4492>.

[RFC5272] Schaad, J. and M. Myers, "Certificate Management over CMS (CMC)", RFC 5272, DOI 10.17487/RFC5272, June 2008, <http://www.rfc-editor.org/info/rfc5272>.

[RFC5751] Ramsdell, B. and S. Turner, "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification", RFC 5751, DOI 10.17487/RFC5751, January 2010, <http://www.rfc-editor.org/info/rfc5751>.

[RFC5958] Turner, S., "Asymmetric Key Packages", RFC 5958, DOI 10.17487/RFC5958, August 2010, <http://www.rfc-editor.org/info/rfc5958>.

[RFC5967] Turner, S., "The application/pkcs10 Media Type", RFC 5967, DOI 10.17487/RFC5967, August 2010, <http://www.rfc-editor.org/info/rfc5967>.

[RFC6090] McGrew, D., Igoe, K., and M. Salter, "Fundamental Elliptic Curve Cryptography Algorithms", RFC 6090, DOI 10.17487/RFC6090, February 2011, <http://www.rfc-editor.org/info/rfc6090>.

[RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, DOI 10.17487/RFC6347, January 2012, <http://www.rfc-editor.org/info/rfc6347>.

[RFC6690] Shelby, Z., "Constrained RESTful Environments (CoRE) Link Format", RFC 6690, DOI 10.17487/RFC6690, August 2012, <http://www.rfc-editor.org/info/rfc6690>.

[RFC7030] Pritikin, M., Ed., Yee, P., Ed., and D. Harkins, Ed., "Enrollment over Secure Transport", RFC 7030, DOI 10.17487/RFC7030, October 2013, <http://www.rfc-editor.org/info/rfc7030>.

[RFC7251] McGrew, D., Bailey, D., Campagna, M., and R. Dugal, "AES-CCM Elliptic Curve Cryptography (ECC) Cipher Suites for TLS", RFC 7251, DOI 10.17487/RFC7251, June 2014, <http://www.rfc-editor.org/info/rfc7251>.

[RFC7252] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", RFC 7252, DOI 10.17487/RFC7252, June 2014, <http://www.rfc-editor.org/info/rfc7252>.

[RFC7959] Bormann, C. and Z. Shelby, Ed., "Block-Wise Transfers in the Constrained Application Protocol (CoAP)", RFC 7959, DOI 10.17487/RFC7959, August 2016, <http://www.rfc-editor.org/info/rfc7959>.

9.2. Informative References

[ieee802.15.4] Institute of Electrical and Electronics Engineers, , "IEEE Standard 802.15.4-2006", 2006.

[RFC4944] Montenegro, G., Kushalnagar, N., Hui, J., and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks", RFC 4944, DOI 10.17487/RFC4944, September 2007, <http://www.rfc-editor.org/info/rfc4944>.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <http://www.rfc-editor.org/info/rfc5246>.

[RFC6838] Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", BCP 13, RFC 6838, DOI 10.17487/RFC6838, January 2013, <http://www.rfc-editor.org/info/rfc6838>.

[RFC7230] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", RFC 7230, DOI 10.17487/RFC7230, June 2014, <http://www.rfc-editor.org/info/rfc7230>.

Appendix A. Operational Scenario Example Messages

This appendix provides detailed examples of the messages using DTLS and BLOCK option Block2. The minimum PMTU is 1280 bytes, which is the example value assumed for the DTLS datagram size. The example block length is taken as 64 which gives an SZX value of 2.

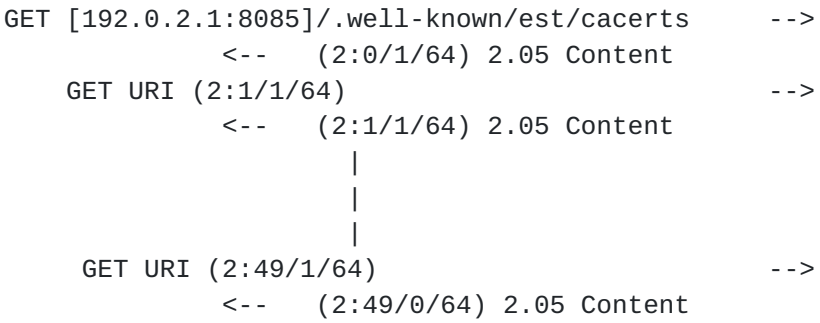
The following is an example of a valid /cacerts exchange.

During the initial DTLS handshake, the client can ignore the optional server-generated "certificate request" and can instead proceed with the CoAP GET request. The content length of the cacerts response in appendix A.1 of [RFC7030] is 4246 bytes using base64. This leads to a length of 3185 bytes in binary. The CoAP message adds around 10 bytes, the DTLS record 29 bytes.

To avoid IP fragmentation, the CoAP block option is used and an MTU of 127 is assumed to stay within one IEEE 802.15.4 packet. To stay below the MTU of 127, the payload is split in 50 packets with a payload of 64 bytes each. Fifty times the client sends an IPv6 packet containing the UDP datagram with the DTLS record that encapsulates the CoAP Request. The server returns an IPv6 packet containing the UDP datagram with the DTLS record that encapsulates the CoAP response.

The CoAP request-response exchange with block option is shown below. Block option is shown in a decomposed way indicating the kind of Block option (2 in this case because used in the response) followed by a colon, and then the block number (NUM), the more bit (M = 0 means last block), and block size exponent (2**(SZX+4)) separated by slashes. The Length 64 is used with SZX= 2 to avoid IP fragmentation.

The CoAP Request is sent with confirmable (CON) option and the content format of the Response is /application/cacerts.



Authors' Addresses

Sandeep S. Kumar
Philips Lighting Research
High Tech Campus 7
Eindhoven 5656 AE
NL

Email: ietf@sandeep.de

Peter van der Stok
Consultant

Email: consultancy@vanderstok.org